

BAB II

TINJAUAN PUSTAKA

DAN LANDASAN TEORI

2.1. Tinjauan Pustaka

Emosi merupakan salah satu aspek yang berperan penting dalam proses pembelajaran. Banyak parameter yang dapat menentukan kondisi emosi seseorang. Wajah merupakan parameter yang paling mudah untuk menentukan kondisi emosi seseorang. Bagian wajah yang dapat menunjukkan kondisi emosi adalah mata, bibir, dan dahi.

Mata dapat digunakan tolak ukur untuk mendapatkan informasi mengenai kondisi emosi. Mata dapat menyampaikan beberapa informasi mengenai apa yang akan dilakukan (Porta, et al., 2012). Pola perubahan mata dapat merepresentasikan emosi yang dialami seseorang dalam suatu waktu tertentu.

Salah satu proses yang dapat dilakukan adalah membuat pendeteksi emosi berdasarkan pola-pola perubahan pada mata. Pendeteksi emosi ini dilakukan untuk mengklasifikasikan emosi berdasarkan pola perubahan mata dalam waktu tertentu. Algoritma *k-Nearest Neighbor* merupakan salah satu algoritma yang dapat digunakan untuk klasifikasi. Klasifikasi yang dapat dilakukan adalah menentukan kondisi emosi seseorang berdasarkan pola perubahan yang terjadi pada pupil mata.

Dalam penelitian *Machine Learning to Differentiate Between Positive and Negative Emotions Using Pupil Diameter* (Babiker, et al., 2015) diketahui bahwa algoritma *k-Nearest Neighbor* digunakan untuk proses pengklasifikasian emosi. Pada penelitian ini pelatihan

data hanya digunakan tingkat konstanta $k=5$. Pada penelitian ini dinilai kurang maksimal karena hanya menggunakan percobaan untuk satu nilai konstanta saja.

Dalam sebuah penelitian *Classification of Emotions in Indonesian Texts Using K-NN Method* (Arfin & Purnama, 2012). Dalam penelitian tersebut mendeteksi emosi berdasarkan pola kalimat yang terbentuk dalam bahasa indonnesia. Dalam penelitian ini menggunakan 7 variasi nilai k dalam penelitian. Dimana dalam penelitiannya nilai *precision* and *recall* terbaik didapatkan dengan menggunakan nilai k pada angka 40. Dalam penelitian tersebut mendapatkan akurasi tertinggi pada nilai $k=40$. Dapat disimpulkan bahwa nilai k berpengaruh besar pada peningkatan akurasi dalam proses pengklasifikasian dengan algoritma *k-Nearest Neighbor*.

Penelitian lain juga melakukan pengujian terhadap pola perubahan pola pupil mata terhadap rangasangan emosi. Pada penelitian yang berjudul *Neural Networks for Emotion Recognition Based on Eye Tracking Data* (Aracena, et al., 2004). Pada penelitian tersebut mendeteksi pola-pola perubahan yang terjadi pada pupil mata dengan menggunakan kelas yang dibentuk berdasarkan data EEG. Penggunaan data pada EEG berfungsi untuk menentukan kondisi yang dialami subjek penelitian. Pada penelitian ini digunakan *pupilometry* sebagai *tools* pengambilan data pola pupil mata.

Musik merupakan salah satu aspek yang dapat merangsang kondisi emosi seseorang. Pada sebuah penelitian yang berjudul *The Eye is Listening: Music-Induced Arousal and Individual Differences Predict*

Pupillary Responses mengatakan respon pupil mata dapat menunjukkan kondisi emosi seseorang. Pada penelitian ini menggunakan 15 subjek wanita sebagai subjek penelitian. Diameter pupil mata akan bereaksi terhadap rangsangan emosi selama 0,2 detik (Aracena, et al., 2004).

2.2. Landasan Teori

Berdasarkan penelitian yang akan dilakukan mengenai "Pendeteksi Emosi melalui pupil mata menggunakan Algoritma *k-Nearest Neighbor*" diperlukan teori-teori yang mendukung penelitian ini. Beberapa teori yang bersangkutan dengan penelitian ini, sebagai berikut :

2.2.1. Emosi

Kata emosi berasal dari bahasa Prancis *emotion*, dari kata *emouvoir*, yang berarti kegembiraan. Selain itu emosi juga berasal dari bahasa Latin *emovere* yang berarti "luar" dan *move* yang berarti "bergerak". Lahey (2003) mengatakan emosi merupakan suatu hal yang dihasilkan oleh fisiologis yang menyebabkan munculnya reaksi emosi. Reaksi ini tidak dapat dibaca namun hanya dapat dilihat dari ekspresinya dan perilaku saja.

Sifat dan intensitas emosi sangat berkaitan erat dengan aktivitas kognitif (berfikir) manusia sebagai hasil persepsi terhadap situasi yang dialaminya. Reaksi manusia terhadap hadirnya emosi, disadari atau tidak memiliki dampak yang bersifat membangun atau merusak. Dengan demikian bisa dikatakan emosi tidak hanya merupakan reaksi terhadap kondisi diri sendiri maupun luar diri sendiri, tetapi juga upaya pencapaian ke arah

pembentukan diri menuju hidup yang transendental (spiritual).

Berdasarkan definisi diatas maka dapat disimpulkan bahwa emosi adalah respon kognitif, perasaan, dan perilaku yang muncul akibat stimulus tertentu. Emosi manusia secara umum terdiri dari 2 bagian sebagai berikut :

a. Emosi Positif

Emosi positif adalah emosi yang mampu menghadirkan perasaan positif terhadap seseorang yang mengalaminya. Hill (2011) mengatakan bahwa terdapat tujuh macam emosi yang masuk dalam emosi positif, diantaranya adalah hasrat, keyakinan, cinta, seks, harapan, romansa dan antusiasme.

b. Emosi Negatif

Emosi negatif merupakan emosi yang selalu identik dengan perasaan tidak menyenangkan dan dapat mengakibatkan perasaan negatif pada orang yang mengalaminya. Pada umumnya, orang yang mengalami emosi negatif cenderung lebih memperhatikan emosi-emosi yang bernilai negatif, seperti sedih, marah, cemas, tersinggung, benci, jijik, prasangka, takut, curiga dan lain sebagainya.

Emosi dapat diekspresikan dalam bentuk verbal atau nonverbal. Ekspresi emosi yang paling mudah dilihat adalah ekspresi nonverbal. Ekspresi nonverbal dapat dilihat dari perubahan raut wajah, dari mata, bibir, dahi, dan pipi. Melalui ekspresi wajah kita dapat melihat seseorang sedang mengalami kondisi emosi tertentu.

2.2.2. k-Nearest Neighbor (kNN)

K-Nearest Neighbor (KNN) termasuk kelompok *instance-based learning*. Algoritma ini juga merupakan salah satu teknik *lazy learning*. KNN dilakukan dengan mencari kelompok *k* objek dalam data *training* yang paling dekat dengan objek pada data baru atau data *testing*. Algoritma *k-Nearest Neighbor* adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. *Nearest Neighbor* adalah pendekatan untuk mencari kasus dengan menghitung kedekatan antara kasus baru dan kasus lama yaitu berdasarkan pada pencocokan bobot dari sejumlah fitur yang ada. Untuk mendefinisikan jarak antara dua titik yaitu titik pada data *training* dan titik pada data *testing* maka digunakan rumus Euclidean.

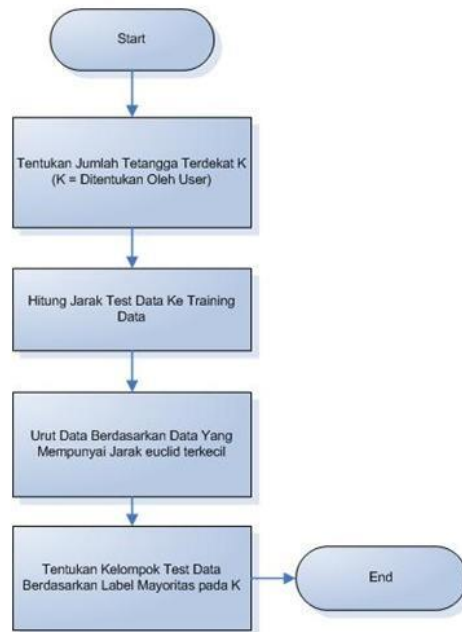
$$D(x, y) = \sqrt{\sum_{k=1}^n (x_k - y_k)^2}$$

Gambar 2.1. Persamaan Euclidian pada k-Nearest Neighbor.

Pada gambar 3.1 merupakan persamaan *Euclidian* pada *k-Nearest Neighbor*. Dapat dijelaskan *D* adalah jarak antara titik *training* yaitu *x* dan titik *testing* yaitu *y* yang akan diklasifikasikan. Pada fase *training*, algoritma ini hanya melakukan penyimpanan vektor-vektor fitur dan klasifikasi data *training sample*. Pada fase klasifikasi, fitur-fitur yang sama dihitung untuk *testing data* (yang klasifikasinya tidak diketahui). Jarak dari vektor baru yang ini

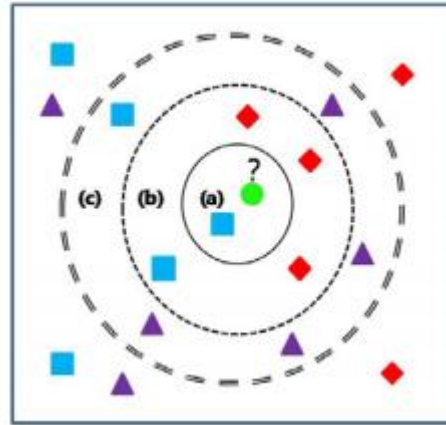
terhadap seluruh vektor training sample dihitung dan sejumlah k buah yang paling dekat diambil.

Nilai k yang terbaik untuk algoritma ini tergantung pada data secara umumnya, nilai k yang tinggi akan mengurangi efek *noise* pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi lebih kabur. Nilai k yang bagus dapat dipilih dengan optimasi parameter, misalnya dengan menggunakan *cross-validation*. Kasus di mana klasifikasi diprediksikan berdasarkan data pembelajaran yang paling dekat (dengan kata lain, $k = 1$) disebut algoritma nearest neighbor. Ketepatan algoritma k -NN ini sangat dipengaruhi oleh ada atau tidaknya fitur-fitur yang tidak relevan, atau jika bobot fitur tersebut tidak setara dengan relevansinya terhadap klasifikasi. Riset terhadap algoritma ini sebagian besar membahas bagaimana memilih dan memberi bobot terhadap fitur, agar performa klasifikasi menjadi lebih baik. Berikut ini adalah diagram flow chart algoritma k -NN :



Gambar 2.2. Langkah Algoritma k-Nearest Neighbor.

Pada proses *k-Nearest Neighbor* proses training dari algoritma ini terdiri dari memasukan vektor dan label dari citra yang akan dilatih. Pada proses klasifikasi titik permintaan yang tidak memiliki label ditugaskan sebagai label k terdekat. Setelah setiap citra dikonversi menjadi bilangan vektor real. Setiap jarak yang paling umum digunakan untuk k-Nearest Neighbor. Berikut ini adalah visualisasi proses klasifikasi menggunakan kNN.



Gambar 2.3. Visualisasi Klasifikasi Menggunakan kNN.

Gambar 2.2 merupakan visualisasi dari kNN dalam mengklasifikasi data. Pada gambar terlihat objek lingkaran berwarna hijau merupakan data *testing* yang akan diuji berdasarkan data *training*. Data *training* merupakan objek persegi berwarna biru, belah ketupat berwarna merah, dan segitiga berwarna ungu. Sedangkan untuk nilai k yang digunakan adalah a , b , dan c . Titik a merupakan hasil klasifikasi dengan k bernilai 1, sesuai dengan konsep dari algoritma *k-Nearest Neighbor* menentukan titik mayoritas jarak terdekat sesuai dengan nilai k yang telah ditentukan, maka dengan nilai k sama dengan 1 data *testing* akan terklasifikasi menjadi objek persegi berwarna biru.

2.2.3. Trainable Weka Segmentation

Trainable Weka Segmentation merupakan plugin yang terdapat pada *Fiji* yang digunakan untuk menyeleksi fitur-fitur pada sebuah citra dengan menggunakan beberapa algoritma. *Trainable Weka Segmentation* ini dapat digunakan untuk mengekstraksi

sebuah citra menjadi data numeris yang dapat diproses dalam Weka. Ekstraksi numeris yang dapat dilakukan *Fiji* adalah mengekstraksi berupa data numeris RGB jika gambar berupa RGB. Sedangkan untuk tipe gambar *grayscale* maka *Fiji* akan mengekstraksi menjadi data numeris berupa *double precision numerical data* (Arganda-Carreras,, et al., 2016). *Trainable Weka Segmentation* memiliki beberapa fitur yang dapat digunakan mengekstraksi data numeris dari sebuah citra. Berikut ini adalah *feature extraction* yang dapat digunakan dalam *Trainable Weka Segmentation*.

a. *Edge Features*.

Edge Features ini berfungsi untuk mengekstraksi fitur garis-garis dalam sebuah citra. Untuk melihat pola garis yang terbentuk dalam sebuah gambar dapat digunakan fitur ini. Didalam *Trainable Weka Segmentation* terdapat beberapa metode yang digunakan untuk melakukan ekstraksi tersebut, yaitu *Laplacian and Sobel filters, difference of Gaussians, Hessian matrix eigenvalues, and Gabor filters*.

b. *Texture Extraction*

Texture Extraction memiliki fungsi untuk mengekstraksi tekstur yang menyusun sebuah citra. Dimana *Texture Extraction* ini dapat digunakan untuk proses pengenalan pola-pola yang terdapat dalam sebuah citra. Didalam *Trainable Weka Segmentation* terdapat beberapa metode yang digunakan untuk melakukan ekstraksi tersebut, yaitu *minimum, maximum, median, mean, variance, entropy, structure tensor*.

c. *Noise Reduction Filter*

Noise Reduction Filter memiliki fungsi untuk mengurangi *noise* dalam sebuah citra. Didalam *Trainable Weka Segmentation* terdapat beberapa metode yang digunakan untuk melakukan ekstraksi tersebut, yaitu *Gaussian blur*, *bilateral filter*, *Anisotropic diffusion*, *Kuwahara*, and *Lipschitz*.

2.2.4. Waikato Environment for Knowledge Analysis (WEKA)

WEKA merupakan perangkat lunak yang menyediakan layanan untuk melakukan pengolahan data dalam *data mining*. Perangkat lunak ini berbasis *open source* dan dibuat menggunakan Java. WEKA dibuat dan dikembangkan oleh Universitas Waikato di Selandia Baru. WEKA merupakan perangkat lunak gratis yang tersedia dibawah *General Public License*. Perangkat ini memiliki fasilitas untuk melakukan *preprocessing data*, *classification*, *regression*, *clustering*, *association rules*, dan *visualiszation*.

Di dalam WEKA terdapat *library* kelas Java yang lengkap untuk melakukan implementasi metode mesin pembelajaran dan algoritma *data mining*. Kemudahan dari aplikasi WEKA ini adalah *library* kelas yang ada dapat digunakan secara bebas pada aplikasi lain. Bahkan, kelas yang ada pada WEKA dapat dijalankan pada aplikasi berbasis web. Hal ini memungkinkan pengguna untuk menerapkan berbagai teknik dan metode *data mining* pada weka untuk data yang ingin mereka olah sendiri tanpa memperhatikan *platform* komputer yang digunakan. (Witten, Frank & Hall, 2011).



Gambar 2.4. User Interface dari Aplikasi Weka

WEKA mulai dikembangkan sejak tahun 1994 dan telah menjadi *software data mining open source* yang paling populer. WEKA mempunyai kelebihan seperti banyak algoritma *data mining* dan *machine learning*, kemudahan dalam penggunaannya, selalu *up-to-date* dengan algoritma-algoritma yang baru. Ian H. Witten merupakan latar belakang dibalik kesuksesan WEKA. Beliau merupakan profesor di *Universitas of Waikato, New Zealand*, yang menekuni *Digital Library, Text Mining, Machine Learning* dan *Information Retrieval*. Pada Weka ada beberapa metode pemilihan *variable* dari suatu *dataset*, diantaranya *BestFirst, ExhaustiveSearch, FCBFSearch, GeneticSearch, GreedyStepwise, RaceSearch, RandomSearch, Rankerdan, RankerSearch*. Metode atau Teknik yang digunakan Weka adalah *Predictive* dan *Descriptive* karena Weka mendukung teknik-teknik *data preprocessing, clustering, classification, regression, visualization*, dan *feature Reduction*. Semua teknik Weka adalah didasarkan pada asumsi

bahwa data tersedia sebagai *flat file* tunggal atau hubungan, dimana setiap titik data digambarkan oleh sejumlah tetap atribut (biasanya, atribut numeric atau nominal, tetapi beberapa jenis atribut lain juga didukung). WEKA memiliki empat jenis *test option* yang dapat digunakan untuk melakukan proses klasifikasi. Proses yang ditangani adalah proses pelatihan dan pengujian. Keempat jenis *test option* tersebut yaitu:

a. *Use Training Set*

Klasifikasi ini akan menggunakan satu data untuk melakukan pelatihan. Kemudian seluruh data yang dilatih sebelumnya juga digunakan untuk proses pengujian.

b. *Supplied test set*

Klasifikasi ini dilakukan evaluasi dengan cara memprediksi seberapa baik satu dataset yang diambil dari sebuah data tertentu yang memang sudah disediakan untuk pengujian. Proses pelatihan akan dilakukan terlebih dahulu menggunakan data latih kemudian proses pengujian akan dilakukan dengan data uji yang berbeda dengan data yang dilatih pada klasifikasi.

c. *Cross Validation*

Klasifikasi ini dilakukan evaluasi dengan *cross-validation* dan menggunakan jumlah *fold* yang tertentu yang dapat diinputkan manual. Pada *cross-validation* akan ada pilihan berapa *fold* yang akan digunakan. Nilai *fold* awal

yang diberikan adalah 10. Proses pengujian akan dilakukan sebanyak nilai fold yang diberikan serta akan dibentuk subset sebanyak nilai fold. Kemudian proses pengujian akan dilakukan menggunakan sebuah subset yang terbentuk dan sisanya akan digunakan untuk proses pelatihannya.

d. *Percentage Split*

Klasifikasi ini dilakukan evaluasi dengan melakukan pembagian data antara data uji dan data latih pada satu dataset dengan menggunakan prosentase. Prosentase yang diinputkan akan digunakan untuk proses pelatihan dan sisanya akan digunakan untuk proses pengujian. Proses ini biasanya dilakukan dengan perbandingan 2/3 data untuk pelatihan dan 1/3 data digunakan untuk proses pengujian.